MODELING OF IMAGE PROCESSING ALGORITHMS FOR HARDWARE-SOFTWARE CO-SIMULATION

IBRAHIM ISA

UNIVERSITI TEKNOLOGI MALAYSIA

MODELING OF IMAGE PROCESSING ALGORITHMS FOR HARDWARE-SOFTWARE CO-SIMULATION

IBRAHIM ISA

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Electrical - Computer and Microelectronics System)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2015

Specially dedicated to My Mother, Father and Siblings.

My enduring and experienced supervisors.

Thank You.

ACKNOWLEDGEMENT

First of all I give thanks to Almighty Allah for giving me the energy and knowledge to come about this work and as well finishing it, verily all praise is due to Him.

I would like to express my sincere gratitude to Prof. Dr. Mohamed khalil Hani and Dr Rabia Bakhteri for their guidance, support and encouragement. Their vast experience and deep understanding of the subject proved to be an immense help to me, and also their profound view-points and extraordinary motivation enlightened me in many ways.

I would also like to express my gratitude to my friends Omid, Hanchien, Vidya, Shansung, Yee Hui and the rest of them at the VeCAD research lab for their help, support and guidance and finally my warmest regards to my family for their encouragement, continuous financial supports and help at all stages of my education.

ABSTRACT

Implementation and verification of algorithms such as image processing algorithms via deploying into field programmable gate arrays can be time consuming and involves a lot of technical complexities. Modern digital systems are expanding in terms of size and design complexity which becomes even more complicated due to task division between hardware and software as well as design and verification teams. Therefore this project proposes to model image processing algorithm such as human skin detection algorithm for hardware-software co-simulation. The skin detection algorithm was first designed as pure software followed by software profiling process to identify the compute-intensive modules. This was followed by the design of hardware accelerators for the compute-intensive modules and hardware-software co-simulation of the whole system.The hardware which is designed using SystemVerilog and the software which is in c programming language communicate through direct programming interface(DPI-C), MATLAB is used as the golden reference model to verify the hardware-software co-simulation. The co-simulation and verification process is automated with the aid of the MATLAB engine. When the hardware-software co-simulation was implemented a speed improvement of up to 2.5 times was obtained as compared to pure software implementation.

ABSTRAK

Pelaksanaan dan pengesahan algoritma seperti algoritma pemprosesan imej melalui menggerakkan ke lapangan tatasusunan get diprogramkan boleh memakan masa dan melibatkan banyak kerumitan teknikal. Sistem digital moden berkembang dari segi saiz dan kerumitan reka bentuk yang menjadi lebih rumit kerana pembahagian tugas antara perkakasan dan perisian serta reka bentuk dan pengesahan pasukan. Oleh itu projek ini bercadang untuk model algoritma pemprosesan imej seperti kulit manusia pengesanan algoritma untuk perkakasan-perisian bersama simulasi. Algoritma pengesanan kulit mula direka sebagai perisian tulen diikuti dengan proses pemprofilan perisian untuk mengenal pasti modul mengira intensif. Ini diikuti dengan reka bentuk pemecut perkakasan untuk modul mengira intensif dan perkakasan-perisian bersama simulasi perkakasan system.The keseluruhan yang direka menggunakan SystemVerilog dan perisian yang ada di c bahasa pengaturcaraan berkomunikasi melalui antara muka pengaturcaraan langsung (DPI -C), MATLAB digunakan sebagai model rujukan emas untuk mengesahkan perkakasan-perisian bersama simulasi. Bersama simulasi dan pengesahan proses adalah automatik dengan bantuan enjin MATLAB. Apabila perkakasan-perisian bersama simulasi dilaksanakan peningkatan kelajuan sehingga 2.5 kali telah diperolehi berbanding dengan pelaksanaan perisian tulen.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

XML	-	Extensible Markup Language
RTL	-	Register transfer level
HW	-	Hardware
SW	-	Software
SV	-	SystemVerilog
DPI	-	Direct Programming Interface
C	-	C Programming Language
CU	-	Control Unit
DU	-	Datapath Unit
FPGA	-	Field-Programmable Gate Arrays
HDL	-	Hardware Description Language
MATLAB	-	Matrix Laboratory
SoC	-	System On Chip
IDE	-	Integrated developement enivironment
DFG	-	Dataflow Graph
DUT	-	Design Under Test
GCC	-	GNU Compiler Collection
IC	-	Integrated Circuit
ISS	-	Instruction Set Simulator
RTL	-	Register Transfer Level

CHAPTER 1

INTRODUCTION

1.1 Background of study

Systems designers are facing two trends in the electronics industry: increasing design complexity and increasing software content in those designs. Hardware modeling or design at register transfer level (RTL) consumes a lot of time and effort. It is impossible to integrate every algorithm(s) into a single piece of hardware. Even if it is done, the end product of the hardware will be costly and not suitable for the market right now. The market now pressures on short design cycle, high fault coverage, high speed and low cost. Since the register transfer level design now is getting more complicated, it is advisable that some of the algorithm should remain in software. Hardware design is done only when performance is taken into consideration. This is due to hardware design can speed up or accelerate the whole system or application [3].

Functional verification has become the biggest bottleneck as it consumes roughly 70% of chip development time and efforts. RTL test benches have become more complex and difficult to manage [4]. New method has to be introduced to reduce the verification cycle.SystemVerilog (SV) can be used to overcome drawbacks, it enhances the design specification method, test-benches language including coverage and assertions application programming interface (API) and direction programming interface (DPI).

Direct programming interface is an interfacing between SystemVerilog and foreign programming language. Usually, the foreign programming language is referred as C or C++ programming language. Designers can call C or C++ code into SystemVerilog or vice-versa by using correct protocol and linking model. Hence by using SV and DPI, designers are able to develop a fast co-simulation environment for their desired applications [4, 5]. In this project MATLAB is used as the golden

reference to verify the hardware and software design.

1.2 Problem Statement

Previously, the hardware and software design are done separately by different designers. One of the ways to verify the system design is using field programmable gate array (FPGA). However verification and implementation of algorithms such as image processing algorithms via deploying into FPGA board can be time consuming and involves a lot of technical complexities.

Modern digital systems are expanding in terms of size and design complexity which becomes even more complicated due to task division between hardware and software as well as design and verification teams.

1.3 Objectives

The goal of this project is to model image processing algorithms for Hardware-Software Co-Simulation. Hence the objectives of this project are as follows:

- To co-simulate the hardware and software parts of the algorithm
- To verify the hardware software co-simulation result output with the MATLAB golden references.
- To automate the simulation process that is the verification and HW/SW co-simulation with the aid of MATLAB engine.

1.4 Scope

- System Verilog HDL is used throughout for the hardware design and C programming language is applied as the foreign programming language
- Free software tools such as Quartus II 13.0 and Altera-Modelsim are used to ensure this work can be repeated and extended in future.

- Case studies that are applied in the HW-SW co-simulation concept and environment are:
  - Image normalization.
  - Skin detection algorithm.
- MATLAB is used as the golden reference for the case studies

1.5 Project Outline

This thesis is organized into six chapters. The first chapter presents the introduction of this work, problem statement, objectives, scope of this project and expected contribution of this work.

Chapter 2 is the background and literature review chapter. It gives some information about HW-SW Co-Simulation techniques. Previous related work on HW-SW Co-simulation; image processing algorithm and skin detection algorithm are presented as well.

Chapter 3 describes the methodology of this work. The HW-SW co-simulation concept and environment are discussed and illustrated as well. Chapter 4 shows the implementation of HW-SW co-simulation concept and environment.

Chapter 5 is the Results and discussion chapter and finally Chapter 6 gives the conclusion of this project as well as the future recommendations.

REFERENCES

1. A. Freitas, “Hardware / Software Co-Verification Using the SystemVerilog DPI,” 2007.
2. D. G. Bailey, *Design for Embedded Image Processing On FPGAS*. John Wiley & Sons, 2011.
3. R. Klein, “Hardware And Software Co-Simulation,” 2000.
4. J. Kong and J. A. K. Kan, “Software And Hardware Co-Simulation Platform For Image Processing,” Master’s thesis, Universiti Teknologi Malaysia, 2014.
5. Sayed Omid Ayat, “Hardware And Software Co-Simulation Platform For Convolution Or Correlation Based Image Processing Algorithms,” Master’s thesis, Universiti Teknologi Malaysia, 2014.
6. J. A. Rowson, “Hardware and Software Co-Simulation,” tech. rep., Redwood Design Automation Inc.
7. M. K. Hani, *RTL Designs of Digita; Systems with Verilog HDL*. Desktop Publisher, 2012.
8. D. V. Rao, S. Patil, N. A. Babu, and V. Muthukumar, “Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages,” *International Journal of Theoretical and Applied Computer Sciences*, vol. 1, no. 1, pp. 9–34, 2006.
9. D. Modi, H. Sitapara, R. Shah, E. Mehul, and P. Engineer, “Integrating MATLAB with Verification HDLs for Functional Verification of Image and Video Processing ASIC,” *Journal of Computer Science*, vol. 2, no. 2, pp. 258–265, 2011.
10. G. Gupta, “Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter,” *International Journal of Soft Computing and Engineering*, vol. 1, no. 5, pp. 304–311, 2011.
11. S. Cho, S. Huh, H. S. Choi, and D. H. Shim, “An Image Processing Algorithm for Detection and Tracking of Aerial Vehicles,” pp. 7482–7487, 2011.
12. J.-E. Ha, “An image processing algorithm for the automatic manipulation of

tie rod," *International Journal of Control, Automation and Systems*, vol. 11, no. 5, pp. 984–990, 2013.

13. T. Statella, P. Pina, and E. A. Da Silva, "Image processing algorithm for the identification of Martian dust devil tracks in MOC and HiRISE images," *Planetary and Space Science*, vol. 70, no. 1, pp. 46–58, 2012.

14. D. Demirovi, "Evaluation of Image Processing Algorithms on ARM Powered Mobile Devices," no. May, 2014.

15. M. Fotouhi, M. Rohban, and S. Kasaei, "Skin detection using contourlet texture analysis," *2009 14th International CSI Computer Conference*, pp. 367–372, 2009.

16. W. R. Tan, C. S. Chan, P. Yogarajah, and J. Condell, "Human Skin Detection," vol. 8, no. 1, pp. 138–147, 2012.

17. Z. Musa, K. Jumari, and N. Zainal, "A method of human skin detection base on background subtraction and color enhancement," *ISBEIA 2011 - 2011 IEEE Symposium on Business, Engineering and Industrial Applications*, no. 16, pp. 498–502, 2011.

18. "Introducing MATLAB Engine - MATLAB & Simulink."

19. Mauromaiorca, Kenfyre, and Et.al, "Normalization (image processing)," 2015.

20. T. Nguyen, "Real-Time Face Detection and Tracking," Master's thesis, Cornell University, 2012.

21. K. Johnson, M. Markowitz, Z. Jones, and Et.al, "YUV," 2011.

22. Martarius, Braksus and Et.al, "Thresholding (image processing)," 2004.

23. H. Jati and D. D. Dominic, "Human skin detection using defined skin region," *Proceedings - International Symposium on Information Technology 2008, ITSim*, vol. 1, pp. 6–9, 2008.

24. "Morphology Fundamentals: Dilation and Erosion - MATLAB & Simulink."